

הסמל

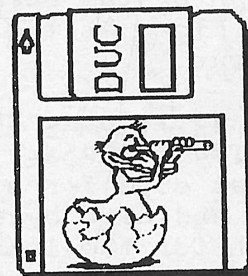
הסמל



D.U.C. 11(E)

ACTION AND REACTION

- * First a rectification. In the new colofon there was a new name Robert, sorry we called you Roland. Please note the correct name in this issue.
- * The NEC-drive does work in combination with the DISCOVERY. So do not let previous articles make you think otherwise!
- * Rob Goedkoop, thanks for the new logo on the front of our magazine. To all other contenders: it was hard to choose one, so we combined a few.
- * Mr. Nagelhout of the Geneology work-groep has recieved reactions from Austria, Germany and Holland. If you too are interested in working out your family-tree on your computer, send him a letter.



I am rather disappointed by the meager reactions to my challenge in the previous magazine, page 29. The only reactions were about RAM-copiers. Come on, ladies and gentlemen, where are you? Anyway, here is a short list of requests still to be fulfilled. If you feel the urge to react, please send me a note so that we don't get five similiar articles for the magazine!

- Manipulation of Ramdisk: enlarge and reduce size, alter sector lengths, give a disc the same format as the RAMdisc (MOVE "d";1 TO "d";5 !!) etc. etc.!!
- Adress 23766 is a familiar adress for microdrive owners. It indicates the last microdrive in use. Is something like this also possible for the DISCOVERY owner with an IC 6116? Are there any other interseting bits of info which we can gather from this?
- We really value your reactions to these queeries!!

BY DICK KRUIHOF

SOME INTERESTING NEWS

By Gert Sleutels

I was very fortunate to be able to rescue this file after it had accidentally been erased. I might have had to type it all again. First, I used LINK II to discover what was still salvageable; the blocks which were still legible were rescued by the LINK option of the same name. I then put the correct codes for a TW2 code block at the beginning of each first block. That works as follows:

```
OPEN #4;1;"name"RND256 (for the first block of the file):
POINT#4,1:FOR a=1 TO 7:INPUT b (the code to put here is the
  first code for a TW2 file: a 3, (= CODE), byte 2 and 3 become
  low and high bytes for a few bytes, and 4 and 5 are low and
  high for the load address (for TW that is 32000 (-> 0 and 125)
  and 6 and 7 are both 0 in this case)
```

```
PRINT#4; CHR$(b); NEXT a
```

Put this code at the correct position at the start of the block
CLOSE #4 and finish everything off.

These alterations were necessary as I could not get hold of the beginning of the article. LINK saved it as a basic program but absolutely refused to save the block altered in LINK. It is of course possible that I had not been doing it according to instructions, but I think my way was not too bad in the end. I did not have the time to check it all out.

So here is more or less what I had written.

ADAPTING TASWORD 2.5

Once again I decided to glue myself behind the keyboard and share the fruits of hours of arduous peering at the listing of TWORD2.5SC. The object was to alter option 7 (printer controls) so that they would work on my STAR printer.

The alterations per word or line are done with the controls put into the text in GRAPHIC mode. The routine is found at lines 1200 - 1221: line 1200 was customised to suit my own needs, just as line 1220 with the names, as they were not the same as Rudie's. Line 1202 sets my printer to print in PICA or ELITE mode with control character 15 or 18. Adding the 3 to the 15 only happens with the correct key-press (CODE t\$=110). Line 1207: change line distance: controlcode 27 and 0 or 2, depending on the outcome of (SGN PI+(i>NOT PI)). This depends on the keys pressed: 1(sgn pi), or 1+1 (if i>0).

TO FLASH OR NOT TO FLASH

I still have not figured out how to include FLASH signs in my listings, the way Rudie has with the default values in his version of TW (2.5). Using the roundabout way of borrowing a FLASH from a previous line and using it for the next, I managed to keep my alterations in style. (EDIT a line, erase all except the flashing cursor and then write your new line.) But that did

not satisfy me. LINK-ed will probably shed some light on the matter, but the trick is still to add the code for FLASH ON/OFF at the correct point; that's just one step too far for me. A text in colour or BRIGHT is easy; it works just like the TRUE and INVERSE VIDEO using number keys for the corresponding colour: Caps and Symbol (E-mode), key for colour (0-6, or 9 for BRIGHT..), type the text and use E-mode at the end again to discontinue. In this way text can be displayed without the commands PAPER and INK. (For eg. on an INPUT line.) If you try this with strings, keep in mind that the string is longer than just the text alone; bytes for colour info must also be taken into account. Do this in a DIM statement, eg. DIM a\$(5,10) instead of DIM a\$(5,6. If you don't, you'll get error reports when printing. Either that, or the colour info is not registered. Whatever happens, the effect is lost.

CAT3PRINT

I have used this type of line in my improved version of Cat3print, as I found it rather messy. I also realised that in condensed mode you could print 3 rows on paper and the paper would fit into the plastic sleeve of the discette. (I still do not use printed disc-stickers.) So here is the altered version with the controls for my printer. It might be necessary to adapt them for use with your printer. I used BETA BASIC 3.0D (see previous DUC's) to turn this listing into a TW file. In the end I still had to move some text about, but it saved me a lot of work. Of course this is not a real discovery; there are already a multitude of ways to achieve this result without a program.

```

0>REM Ger Sleutels          Cat 3 wide on the printer
10 CLEAR #: OPEN #3;"b": LPRINT CHR$ 27;CHR$.15: CLOSE #3:
  REM condensed print on STAR SG-10 in IBM-mode
20 CLS #: BEEP .05,10: LET drive=0: LET side=1
30 PRINT AT 21,5;"Which drive :"; FLASH 1;"1"; FLASH 0;"/2":
  PAUSE 0: LET d$=INKEY$
40 IF d$=CHR$ 13 THEN LET d$="1"
50 LET drive=CODE d$-48: IF drive<1 OR drive>2 THEN GO TO 30
60 PRINT AT 21,5; FLASH 1;"L"; FLASH 0;"Left/Middle/Right"
70 PAUSE 0: LET k$=INKEY$: LET side=1
80 IF k$="m" OR k$="M" THEN LET side=2
90 IF k$="r" OR k$="R" THEN LET side=3
99 REM left print default
100 GO SUB 200*side
109 REM printing
110 OPEN #3;"t": CAT #3;drive: CLOSE #3
120 CLS #: PRINT AT 21,5;"AGAIN ? Y/"; FLASH 1;"N";FLASH0
130 PAUSE 0: LET j$=INKEY$
140 IF j$<>"y" AND j$<>"Y" THEN GO TO 190: REM margin
  at standard posi
150 GO TO 60
190 CLEAR #: OPEN #3;"b": LPRINT CHR$ 27;CHR$64 :CLOSE #3:
  REM reset printer
199 REM Left margin at 10
200 OPEN #3;"b ": LPRINT CHR$27;"1";CHR$ 10
210 CLOSE #3: RETURN
399 REM Left margin at 50
400 OPEN #3;"b ": LPRINT CHR$ 27;"1";CHR$ 50

```



```
410 CLOSE #3: RETURN 599 REM margin at 90
600 OPEN #3;"b ": LPRINT CHR$ 27;"1";CHR$ 90
610 CLOSE #3: RETURN
999 CLEAR: SAVE *PI/PI;"*CAT3print" LINE 1000 : STOP
1000 CLEAR #:LOAD *1;"cat3code"CODE 8*1024: RUN
```

VARIA

Some remarks regarding:

RND access-files: the End of File function USR 432 does work with sequential files, as we can gather from the 'manual'. It says that when reading a disc, the drive keeps passing PRINT#4, to intercept the end. If I can figure out if and how this works with RND-access I will certainly let you know. I have been working on that, but sometimes you just have so much to do and no time to do it in.

CALENDAR PROGRAM:

The calendar program mentioned before is now complete. For those who like the program when they see it, I have some remarks for customizing the program. The public-holiday array is not completely full, and can be extended to encompass all the family's birthdays, etc. Same goes for the numerical arrays K, F and T, which serve to sort the holidays. My original idea, to use two different streams for printing, has been abandoned. For the print on screen I simply ~~repeat~~ the print routine with altered variables (the screen is smaller than paper).

VIDEO-AMPLIFIERS AND KEYBOARD-INTERFACES:

This is a reaction to Ton's article about a video amplifier. I connected my monitor using the direct-soldering method, and now wonder whether an amplifier would have given better results.

Despite my lack of electronic knowledge I dared to do some soldering in my machine; I connected a RESET, moved the print from the DK TRONICS to the keyboard -which is now connected to the DISCOVERY using ribbon cable- so that when I finally decide to purchase a 'real' keyboard connecting that will be a piece of cake. I had read about such a connection in a magazine, and had consulted a local electronics supplier. He thought buffering the signals from the keyboard to the computer would not be necessary using a cable of only 50cm. long. I used less, without buffering, and all seems to be working well.

VTX-HASSLES:

I write this two days later, after having tried unsuccessfully to transmit my text through the modem. That has lead to the following remarks. I've been messing about with my 'phones, as I got very irritated by the fact that the other phone rings while I dial on my 'modem-phone'. I managed to cut that out, but the

THE PROGRAM BANK

By Ton A1

The following articles are now for sale through the program bank. Remember to order as stated in the colofon.

DUCDISC-1 (all-in).	Hfl. 10,00
DUCDISC-2 (all-in).	Hfl. 12,50
MOUSE-UTILITIES (all-in).	Hfl. 25,00
LINK-II (20 K).	Hfl. 12,50
LINK-ED (20 K).	Hfl. 10,00
LINK-II + LINK-ED (40 K).	Hfl. 20,00
PLOMP's UTILITIES (110 K)	Hfl. 10,00
Dito with printed manual.	Hfl. 12,50
TASWORD 2.5 (30 K).	Hfl. 5,00
M/C-runprogram (ULTI-MATE) (20 K)	Hfl. 2,50
CONVERSIONROG. for EXT. ART STUDIO (10 K)	Hfl. 10,00
Dito for PASCAL 1.6M (10 K)	Hfl. 10,00
Dito for The Quill (5 K)	Hfl. 5,00
Dito for The Illustrator (5 K)	Hfl. 5,00
GRAFICA (50 K).	Hfl. 5,00
SECTION IV (55K)	Hfl. 2,50
SCIUC Public Domain Software (all-in).	Hfl. 10,00
TASPAS/PASTAS (20 K).	Hfl. 5,00
NEW: BASICODE-3 (30 K)	Hfl. 2,50

N.B: Hfl = dutch guilders!!!

With "all-in" prices there are no extra costs for you.

All other prices are for the program only, so please add Hfl. 10 per disc to the price. In the table you can see the length of the programs, and work out how many discs you will need for your order. Please remember that we only work with discs formatted at 178k. We cannot (at this point) cope for other formats.



result was that I could not replace the horn while on-line. That seems okay for direct connection with another terminal, but wrecks havoc with DATA transport.

Is there anyone who can shed some light on this subject, i.e how to connect your phone to the modem in such a way that after making contact, you can simply hang up the phone without interrupting the connection?

Ger Sleutels
A.v.Hensbeeksingel 148
2803 LK GOUDA
The Netherlands

sinclair

07

impuls

Software, games

Golf	fl 32,-
1942 (arcade)	36,-
Konami's Coin-op Hits (o.a. Green Beret, PingPong)	39,-
Academy	36,-
Star Gilder (ook 128 K)	59,-
Fairlight II	39,-
Super Soccer (voetbal)	32,-
Lighthforce	32,-
Gauntlet	36,-
Sam Fox strippoker	36,-
Psi Chess 3-D	39,-
Nemesis (nieuw!)	32,-
Way of the exploding Fist II	32,-
Enduro Racer (levens-echt)	29,-
Sigma 7	35,-
Bomb Jack II (succes verzekerd)	29,-
Artist II, tekenprogramma	59,-
Ace of Aces, flight simulator	36,-
Shadow Skimmer	29,-
Shockway Rider	32,-
Arkanoid, de hit van '87	29,-

Aanbieding 1



fl 399,-

Midi interface
48/128/+Z..... fl 125,-

Software, serieus

Tasword III, cartr.	fl 69,-
Tascopy, screendump	45,-
Tasprint, 6 fonts	45,-
Tas-diary, dagboek	45,-
Mini-Office	29,-
Omnicalc	69,-
Skip-64, 64 kolom	19,-
Art Studio	65,-
Masterfile	69,-
Beta Basic	69,-
Laser Genius	65,-
Laser Basic	65,-
Laser Compiler	65,-

Aanbieding 2

DISCIPLE

Disk-interface voor 48 - 128k Spectrum, incl. printer-, joystick- en net- workpoorten. Ook met Snap-shot- button voor wegsaven naar disk van één programma. (48K programma's, laden binnen 4 sec.!!)	fl 345,-
Cumana Disk-interface voor Sinclair Q.L.	fl 295,-
3 1/2 - diskdrives 1 Mbyte, Compleet vanaf	fl 500,-

Aanbieding 3



ZX-Spectrum Plus Twee

fl 499,-

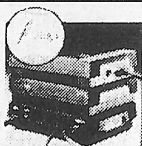
Software, 128 K.

Art Studio 128 k	89,-
Star Glider	59,-
Winter Games	39,-
Knight Time	19,-
3-weeks in paradise	29,-
Music Box	45,-
Tasword 128	65,-
Zub	19,-
Glider Rider	39,-
Samanta Fox Strip	39,-
Rasputin	36,-
Fairlight	36,-
Thanatos	39,-
Gladiator	36,-
Laser Genius (Assembl.)	65,-
Anal of Rome	39,-

Opusdisk Software

Grootboek Opus. Nederlands met 600
mutaties p.p. incl. kas, bank en giro-
boek. Alleen voor 80-koloms printers.
(look op cartridge verkrijgbaar) . . . 89,-
Tascopy Opus 45,-
Masterfile 69,-
Business Pack 59,-
Tasword III, the final
wordprocessor 79,-

Aanbieding 4



Tandata

fl 360,-

Compleet modern
pakket voor Sinclair
Q.L. Incl. auto-dial en
auto-answer.
Elke Q.L-bezitter start
nu z'n eigen databank
voor maar

Printers

Centronics G.L.P. een 80-koloms
printer voor normaal papier. Met
ser. en par. interface. Div. letter-
typen o.a. N.L.Q. 445,-
Citizen 120 D, schitterende par.
printer met vele mogelijkheden, o.a.
proportioneel, inversie, N.L.Q. . . 795,-

Monitors

R.G.B.-monitor (kleur) voor 128k of
Spectrum + 2. Zeer scherp, zeer fraai,
zeer goedkoop 695,-
Monochrome vanaf 229,-

Diversen

Wij leveren software-op-maat. Zowel
op Spectrum, Sinclair Q.L. alsook op
IBM-compatibles.

Data-Skip levert ook de Amstrad
PC1512 tegen soepele prijzen. (bijv.
2 x 360K, monochrome 2190,-
ex. BTW.

Sinclair Q.L. hardware

Sinclair Q.L. compleet	399,-
CST Disk-interface	345,-
Modem-pakket, compleet	360,-
NLQ-printers vanaf	445,-
Monitors vanaf	229,-

Aanbieding 5



AMX-muis, incl. software

fl 249,-

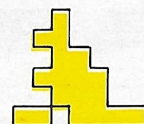
ZX-Spectrum Hardware

Multiface One	175,-
Multiface One 128	199,-
Videoface Digitiser	249,-
Cartridge box	19,-
VTX-5000 modem	149,-
ZXL-printinterface	199,-
3 1/2" diskette	4,-
Konix Speedking, joystick	39,-
Joystick Interface	49,-
Saga Elite, toetsenbord	275,-
Saga TWO-plus	225,-
Opus Discovery	495,-

POSTORDER: BEL 01820-20581
porto software fl 2,50
hardware fl 5,-
rembours fl 10,-

Prijswijzigingen voorbehouden
CST Thor bel voor inlichtingen
en prijzen

Bestellingen d.m.v. ingesloten girobetaalkaart of voor-
uitbetaling op giro 4727958 t.n.v. Data-Skip, Gouda.
Of kom langs in onze winkel. Di. t/m za. van 10-17 uur.



REACTIONS TO DUCDISC 2

(Ed:

Of course the main complaint is the fact that the programs and text files have not been translated. We are very much aware that outside of Holland not many people speak dutch. It is for that reason that we found someone to translate the magazine. The poor guy has been working so hard on that, that only now has he been able to get down to translating the discs. Ducdisc 1 has now been fully translated; those who have recieved the dutch version and really cannot do anything with it, are invited to send the disc back to the program library for a free replacement in the English language. Watch this magazine for translated versions of the other Ducdiscs.)

On DUCDISC-2 you find two programs written by me: SCRABBLE and CHARSTW2/3, together with the code block clear 43180 belonging to the program SCRABBLE. Some remarks about these:

clear43180 and SCRABBLE:

The manual on DUCDISC 2 says that the code block "clear43180" contains DEF KEYS from BETA BASIC 3.0 and that you can look at this by LIST DEF KEY. That is correct. However, you will get a better listing of the DEF KEYS by depressing defined key 'c'.

Besides these DEF KEYS which take up addresses 43180 to 44253, the code block also contains four other things:

At addresses 44254 to 45603 we find the bytes of the H\$ (LET H\$=MEMORY\$(44254 TO 45603)). This H\$ is in fact the layout of the SCRABBLE board.

At addresses 45604 to 46933 we find the routine from the prog. SPEEDYLOAD, to SAVE and LOAD to/from tape at double speed. You should: PRINT USR 45604: SAVE, LOAD, MERGE or VERIFY etc.

At adress 46934 to 46962 is the routine to find the adress of a REM statement in a program line. You should DPOKE the number of the program line to adress 23296 and then key: Let a=USR 46934: PRINT a. The screen will show the adress of the byte following this REM. If the REM is at the start of the line, you find the starting adress of the program line by subtracting 5 from a. This routine can be found in the book: ZX-SPECTRUM machinecode-routines by J. Hardman and A. Hewson.

At addresses 46963 to 47041 a routine from the same book for compacting programs, by removing spaces, etc.: RANDOMIZE USR 46963.

The CODE-block clear 43180 was written in BETA BASIC 3.0 before the DISCOVERY version. You must therefor use that version to utilize these routines. BB 3.0(D)'s machine code has a much lower starting adress: 46830 instead of 47071. If you first LOAD code block clear 43180 and then BB3d (NOT THE OTHER WAY ROUND!!), the machine code routines in the block of BB 3.0(D) will be overwritten and therefor useless. Fortunately the program SCRABBLE is not interrupted by all this.

The routines "adres REM" and "comprimer" are completely relocatable, but the fast load routine isn't. If you want to use the former two routines with BB3D you should alter the CLEAR address in line 2065 from 43180 to 42939 and load the code block as follows: LOAD *1;"clear43180"CODE 42940. The routines are now 241 bytes lower, as are the call addresses. These numbers occur again in line 7080 of the 'run' program. Alter them there too. Don't forget to lower the number 45604 in line 7101 to 45363. Also alter the following in line 110 of the "SCRABBLE" program: LET H\$=MEMORY\$(43213 TO 45362).

If you want to use the quickload routine, it should be placed at an address which is 1024 bytes lower than that of the old routine and then of course lower everything 1024 bytes instead of 241 bytes, leaving you with 783 bytes between these routines and BB 3.0D. The quickload routine needs to be altered as it is not completely relocatable. POKE the following at the corresponding addresses:

```
44646,175 44684,174 44706,174 44722,174 44750,174 44766,174 4481
9,175 44834,175 44841,175 44856,175 44866,175 44913,175 44924,17
5 44938,175 45124,176 45317,179 45333,174 45418,177 45423,178 45
481,174 45591,177 45658,177 45681,177 45714,178 45774,178 45873,
174 45893,174.
```

In one of the SPECTRUM magazines (I don't remember which) I saw a routine which could be loaded into BB 3.0's machine code at the address where function FN c() or COSE() is called. Line 0 also needs some alteration. In my new version I put this line as "ALTER line 0" and the corresponding mc under the name "adres LINE". You first load BETA BASIC and then: MERGE "ALTER line 0" and LOAD "adres LINE"CODE. (ALTER and LINE are one-key words!) Now key: PRINT COSE(n) so as not to print the COSINE of angle n, but the starting address of line n! Recently I made 7 stringarrays under the names a\$() to g\$(). Using BETA BASIC I tried to delete the strings out of each array which also appeared in other arrays. Using function INARRY() and the new COSE() this worked a treat. To illustrate this, here's a good listing:

```
10> FOR g=97 TO 103
    LET x=DPEEK(23621)          , i.e. current line number
    BORDER g-97
    POKE COSE(x+30)+5,g
    POKE COSE(x+60)+23,g      Was saved as
20 FOR h=1 TO 600              g=103, so that we now see in
30 POKE 23692,255             lines 40 and 70: g$.
40 PRINT g$(h);" ";          i was 102, so that we now see in
50 DIM p(7)                   line 70: f$
    FOR i=97 TO 103
        IF i=g THEN GO TO 140
60 POKE COSE(DPEEK(23621)+10)+11,i
70 LET iy=INARRAY(f$(1),g$(h))
80 IF iy THEN LET p(i-96)=iy
    LET p(g-96)=h
90 NEXT i
100 IF NOT p(g-96) THEN GO TO 150
```



```

110 LET h=h-1
    FOR a=1 TO 7
        POKE COSE(DPEEK(23621)+10)+11,a+96
120 IF p(a) THEN DELETE g$(p(a))
        PRINT INVERSE 1;">";a;" ";
130 NEXT a
    GO TO 150
140 NEXT i
150 PRINT
    NEXT h
    PRINT ''
160 NEXT g
170 PRINT '''
    FOR a=97 TO 103
        PRINT CHR$ a;"$ ";LENGTH(1,CHR$ a+"$")
    NEXT a

```

This article was written in TASWORD II. The previous listing has not been added to it. It is simply part of the text file. Changing a listing into a text file using BETA BASIC goes as follows:

```

1> CLOSE #4
    OPEN #4;"M";1;" LIST #4" OUT
    LIST #4;10 TO
    LET part=0
    CLOSE #4
    KEYIN "9 REM "
    DELETE 9 TO
2 CLEAR 27999
    CLOSE #4
    OPEN #4;"M";1;" LIST #4" IN
    LET P=28000
3 DO UNTIL EOF(4)
4 INPUT #4; LINE A$
    LET B$=SHIFT$(7,A$)
    LET AB=MOD(LEN B$,64)
    POKE 23692,255
    PRINT B$
    IF AB THEN LET B$=B$+STRING$(64-AB," ")
5 IF P+LEN B$>=46830 THEN LET part=part+1
    SAVETWFILE
    LET P=28000
6 POKE P,B$
    LET P=P+LEN B$
    LOOP
    SAVETWFILE
7 DEF PROC SAVETWFILE
    INPUT "NAAM: ";("part" + STR$ part+" " AND part);T$
    SAVE T$CODE 28000,P-28000
    VERIFY T$CODE
    END PROC
8 DEF PROC SAVE
    CLEAR
    SAVE " LIST ->TW"
    VERIFY " LIST ->TW"
    STOP
    END PROC

```


The listing can now be simply merged into a text file.

If you send me your disc, I will add two other great games to the two versions of SCRABBLE: YAHTZEE and MAH-YONG. I will also be sending these programs to Ton Al, so that in time these will be appearing on one of the DUCDISCs.

CHARSTW2/3

You have been misinformed in the explanatory text on DUCDISC 2. The true reason why you have to qualify the character when LOADING and SAVING is the following. An example: let's say you want to define the German sz as being GRAPHICS 2. You would call the B now. You would then add the tail and save it as GRAPHICS 2. This way you avoid having to design new characters altogether. You simply use existing, almost similar characters. If you do need to design a completely new character, you use a SPACE (code 32).

EDUCATION

On DUCDISC-2 you will find an administration program by Rob Hubbrechtsen. I too have written one, based upon TASWORD II. TASWORD II's text bytes are directly accessible with PEEK and POKE, which offers us the possibility to transform this word processor into a very adult spreadsheet program. In this way I made a program which helps me in my work as a teacher. If you're interested, please contact me.

A.J. Schoneveld
Roosevelttlaan 4
3844 AJ HARDERWIJK
The Netherlands
Tel. 03410-14601

=====

SOFTWARE REVIEW (BRADWAY)

By Ton At

In DUC-Magazine no. 10 we reviewed software from BRADWAY, for which you will find discount coupons in this magazine. To illustrate completely what you can get from BRADWAY, I will now review the rest of the programs, so that you know what you're getting when you order Letta-Head plus and Dumpy-3 from them at the special DUC discount.

I have been using LETTAHEAD+ for some time now, and the result can be seen if you order discs from the library: I do the lables with this great program!! It is a program with which you can design letterheads and lables and then print them in any amount you like.

The programma is not aimed specifically at OPUS Discovery users, but it goes a long way in supporting all the extra possibilities. Things are even made easy by the fact that when choosing printer interfaces, the DISCOVERY is one of the options. All SAVE, LOAD and CAT functions are supported.

The program's strongest point is the fact that you can design accurately to the pixel! All you do is define a window and then scroll that window any way you want using the cursor keys.

You can also inverse the window or even erase it. The program is very flexible on these points. You can also input text. The program has a number of standard character sets. Text can be printed proportionally. This part is a bit like a word-processor, only the characters are displayed very slowly. So working on text asks a little patience. It is not really a big problem, as the amount of text on a lable or letterhead is limited anyway.

The program also has some possibilities for drawing, like PLOT, DRAW, CIRCLE and FILL. It does not pretend to be an advanced graphic program, but it certainly goes in the right direction: any drawing in the format of a SCREEN\$ can be loaded and used in your design.

The program has two 'modes' in which to draw; either letterhead or lable. In the latter mode you have the option of a number of lable formats or you can define the size yourself. You will be presented with an outline representing the lable. (There is even a standard cassette lable.) The other mode is the letterhead option which uses the complete width of the screen.

For printing your design you again have the two options letterhead (heading) or lables. The program requires some information from you: how many prints? How wide? How high? From which TAB position to start printing? Which FORMFEED should be given, etc. etc. When printing lables it also asks how many are to be printed alongside each other and the size of the space between. And then it starts printing!

At this point it is important to know exactly what your printer can and can't do. I must admit that the program can cope with

most printers, but when you place your order, specify which printer you have, so that BRADWAY can make sure your program is customized to suit your needs. Once you really understand the program, you will find it a boon to all those things which you print regularly.

- - -

DUMPY-3 is a program which helps you create screen dump programs and that really copes for a lot of printers and interfaces. But again, it won't hurt to mention your printer type when ordering.

The program starts by asking which interface you use and of course offers OPUS as an option. Then follow questions about the size of the screendump and whether you want the dump on its side or not. It also asks the TAB position from which to start printing.

It is a long list of questions, as the program also needs to know which part of the screen needs to be printed. The SPECTRUM has 32 chars on a line, by 24 lines per screen. These are the sizes the program uses. You input line- and char position, width and height and TAB start, and the program does it all for you. Once oyu've got that all done you can even make a test print and if necessary correct some info. You then SAVE the customized version of your very own, personalized screendump routine. Oh, by the way, you can even tell the program into which part of memory it is to be loaded. It shows you how many bytes your routine takes up and if that is 256 or less, you could even pop it into the printerbuffer. Call it when you need it with a simple RANDOMIZE USR 23296.

Practise has taught that these programs really do work with most printers. However, to avoid disappointment, I must stress that you state your printer when ordering!!! If your printer really won't work with the program, your money will be refunded by BRADWAY. In that case you may, as far as I'm concerned, consider yourself the proud owner of a printer which is "totally-incompatible-with-everything".

One more thing about BRADWAY's programs. In the previous mag you read about the DISC MANAGER and ASTRUM+. In this mag I told you about LETTAHEAD and DUMPY. In the next issue of our magazine I will review LIN-O-TYP.

=====

By Jan Dubois

Reading through my DUC's, I come to the conclusion that a lot of people have trouble getting the programs "VU-Calc" and "VU-File" to work on the Opus Discovery. Well, here is the solution, step by step.

- VU-CALC: 1) Clear memory. Type MERGE "" and start the tape. This LOADs the BASIC, without RUNning it;
 2) Remove line 10 and change lines 1000 and 4000 by adding a *1;
 3) In line 1000, after CODE, add address 30457.
 i.e..... CODE 30457
 4) SAVE the BASIC by: SAVE *1;"VUCALCB" LINE 3200
 5) N.B. Alter all values XXXXX in the whole listing by VAL"XXXXX"! This saves memory space.
 6) Type in LOAD"c" CODE 25232 to load the main part of the machine code.
 SAVE this by: SAVE *1;"VUCALCC" CODE25232,5270
 7) Make a 'loader': 10 LOAD *1;"VUCALCC" CODE 25232:
 LOAD *1;"VUCALCB"
 SAVE it by: SAVE *1;"VU-Calc" LINE 10

If you follow these steps, VU-CALC will present you no more problems.

A similiar thing can be done for VU-FILE.

- VU-FILE: 1) Clear the memory.
 Type in: MERGE "VUFILE" and start the tape.
 2) After the BASIC-part has loaded you will see error report O.K. Stop the tape. Remove line 50. Remove INK 7 in line 100.
 3) Complete all SAVE, LOAD and VERIFY commands by adding *1; Replace all values by VALs.
 SAVE the altered BASIC: SAVE *1;"VUFILEB"LINE 100
 4) Clear memory.
 LOAD machinecode from the tape by.:
 LOAD "C"CODE 25088
 and SAVE it to disc by:
 SAVE *1;"VUFILEC"CODE25088,5640
 5) Make a 'loader': 10 LOAD *1;"VUFILEC" CODE 25088:
 LOAD *1;"VUFILEB"
 SAVE it by: SAVE *1;"VU-File" LINE 10

Jan Dubois

=====

Proud users of TASWORD III will know that it is possible to transform their old T2 texts into the new T3 format using a program on the T3 disc; T2T3. Many people have asked us (each for their own reasons!) how to invert this procedure. Well, one of our members came up with the answer. Just goes to prove that not everyone is frightened off by the challenges we offer. The following listing can be added to the existing TASWORD 2, but do watch the line numbers, and alter accordingly if necessary.

```

2010 INPUT "Put the disc with TASWORD III text into the drive
      and press ENTER ";a$
2020 CAT SGN PI
2030 INPUT "Name file: ";t$"From which line? ";r
2040 LET p=VAL "31999+64*r"
      CLOSE #VAL "4"
      OPEN #VAL "4";"m";SGN PI;t$IN
2050 PRINT #VAL "4";
2060 LET a$=INKEY$#4
      IF a$=CHR$ VAL "13" THEN PRINT
      GO SUB VAL "2090"
      GO TO VAL "2060"
2070 IF a$=CHR$ VAL "10" THEN GO TO VAL "2060"
2075 IF a$="" THEN PRINT "END"
      CLOSE #4 : STOP
2080 LET p=p+SGN PI
      IF p<VAL "52480" THEN PRINT a$;
      POKE p,CODE a$
      GO TO VAL "2060"
2085 PRINT "FILE FULL!": STOP
2090 LET rest=p-VAL "64*INT (p/64)"
      LET p=p+VAL "64-rest-1+(64 AND rest<>63)"
      POKE VAL "23692",CODE " COPY "
      PRINT INVERSE SGN PI;VAL "(p-31999)/64"
      RETURN
2095 CLEAR
      SAVE *1;"TW III->II"
2096 VERIFY *1;"TW III->II": STOP

```

The memory-saving technique of representing a value in a program by eg. SGN PI for 1 and VAL "number" for a number, need not necessarily be executed manually. Just write the program in the normal way and then let the following BETA BASIC program loose on it.

```

1 LET a=COSE(10)-1
  LET start=a
  LET basic=DPEEK(23627)-a
2 LET ab=DPEEK(a+3)
  LET rn=PEEK (a+2)+256*PEEK (a+1)
  LET a$=USING$("####",rn)
  PRINT PAPER 5;a$;
  FOR k=a+5 TO a+ab+3
3   LET p=0
   LET pk=PEEK k
   IF pk=13 THEN GO TO 6
   ELSE IF pk<>14 THEN GO TO 5

```



```

ELSE FOR n=k-1 TO k-13 STEP -1
  IF (PEEK n>=48 AND PEEK n<=57) OR PEEK n=CODE "." OR
    PEEK n=CODE "e" OR PEEK n=CODE "E" THEN LET p=p+1
NEXT n
4 IF p=0 THEN GO TO 5
ELSE LET a$=a$( TO LEN a$-p)
LET b$=CHR$ PEEK (k-1)
IF p=1 AND (b$<="3" AND b$>"2") THEN LET a$=a$+("NOT PI"
AND b$="0")+("SGN PI" AND b$="1")+("INT PI" AND b$="3")
LET k=k+6
GO TO 3
ELSE LET m$=MEMORY$(k-p TO k-1)
LET a$=a$+"VAL "+CHR$ 34+m$+CHR$ 34
LET k=k+6
GO TO 3
5 LET a$=a$+CHR$ pk
PRINT PAPER 5;(CHR$ pk AND pk>=32);
NEXT k
6 PRINT
ON ERROR 7
  BORDER 7
  KEYIN a$
  POKE 23692,255
  LIST rn TO rn
  BORDER 7
LET a=a+DPEEK(a+3)+4
IF PEEK a<>13 THEN BORDER 0
STOP
ELSE IF a>=DPEEK(23627)-1 THEN GO TO 8
ELSE GO TO 2
7 BEEP .1,12
IF lino=6 THEN BORDER 1
LET a$=a$( TO 4)+" REM "+a$(5 TO )
POP
GO TO 6
ELSE PRINT lino;" ";stat;" ";error
STOP
8 ON ERROR 0: LET valbasic=DPEEK(23627)-start : CLS
PRINT basic;" -> ";valbasic'INT (10000*(basic-valbasic)/
basic+.5)/100;" % saved""Press any key"
PAUSE 0 : CLS
FOR n=10 TO 9999
  CLS : LIST n TO n
  LET a$=""
  FOR x=0 TO 3 : LET a$=a$+SCRN$(0,x)
  NEXT x
  LET n=VAL a$: STOP
NEXT n : STOP
9 DEF PROC SAVE
CLEAR: SAVE "comprimer"
VERIFY "comprimer" : STOP
END PROC

```

A.J. Schoneveld
Roosevelttlaan 4
3844 AJ HARDERWIJK
Tel. 03410-14601

As was expected, I recieved many reactions to my article on cracking, some DUC's ago. The almost daily qustions were about intricate cracking and "headerless-cracking". Well, that is what this article is about.

REMEMBER: EACH COMMERCIAL PROGRAM IS PROTECTED BY COPYRIGHT.

(If you have spent many arduous hours, days or months in perfectioning a program, you will understand the the disasterousness (?) of a professional discovering his own program cracked by clever tricks. Nothing is fair in love and war, and dishonesty remains the same, be it twenty pence or twenty pounds' worth. ED.)

This is an example of a standard BASIC-loader:

```
10 CLEAR 29999: INK 0: BORDER 0: PAPER 0: CLS
20 LOAD "" CODE: RANDOMIZE USR 60000
30 REM STANDARD BASIC LOADER
```

It is obvious that this program lowers RAMTOP to 29999 and then loads the code ans starts it. Practise has proven that from this point on there is no return to BASIC. In other words, LOADING this to SAVE it at a later stage is impossible. Despite all this, we are still going to try and transfer a commercial program from tape to disc without fancy gadgets like MULTIFACE 1

LOAD a HEADERREADER to see where the code gets placed and what the length of the routine is. In my example, I have loaded a code at 60000 with a length of 100 bytes.

CLEAR the computer and load your favourite disassembler.
Load the code from tape and make a hardcopy to the printer.
Our example would look like this:

```
DI                ; switch off interrupts
SCF               ; load machine code
LD IX, 16384      at adress 16384 and
LD DE, 6912       length of 6912
LD A, 255         bytes. (i.e.
CALL 1366         a SCREEN$.
SCF               ; again load machine code
LD IX, 30000      this time at 30000
LD DE, 29999      with a length of 29999
LD A, 255         bytes.
SCF
CALL 1366
EI                ; switch interrupts on again.
CALL 30000        ; start main program.
```

Seems simple enough. The list tells me that first a screen is loaded, and only then does the main program get loaded. And of course started.

A remark at this point. If the A register contains any other value than 255, then discovering Mcode is a fine art!!! So, on to the next part of the disassembly.

How do we get this all to disc? Simple. Replace the last CALL 30000 by RET, which says as much as 'GO BACK TO BASIC'. Now save the code to disc with a small BASIC loader.

```
10 CLEAR 29999: LOAD *1;"test" CODE 60000
20 RANDOMIZE USR 60000
30 SAVE *1;"screen" SCREEN$
40 SAVE *1;"code " CODE 30000, 29999
```

Now start the BASIC-program. STOP the cassette once the screen has loaded, to save having to rewind it once the screen has been saved to disc. Once that is done, you make a new BASIC-loader:

```
10 CLEAR 29999
20 LOAD *1;"screen"SCREEN$
30 LOAD *1;"code "CODE
40 RANDOMIZE USR 30000
50 REM NEW BASIC LOADER
```

Well, that was simple enough.

Sometimes the machine code is in a BASIC line. To grasp this, let us look at how to get a routine into BASIC. We do the following:

```
10 LET start=PEEK 23637+256*PEEK 23638+5
20 REM 12345678901234567: REM reserve memory bytes
30 RESTORE 40: FOR f= start TO start+17: READ a:
  POKE f,a: NEXT f
40 DATA 33,64,156,17,192,93,1,27,33,237,176,49,191,33,205,192,
  93: REM This is the routine in decimal values.
50 RANDOMIZE USR start
```

So what actually happens? We search for the basic-pointer in the system variables. Next we POKE the DATA from line 40 into the REM statement in line 20. This can be restarted by the command in line 50.

So how will that look, later?

To us it will look like a mess with reports like "Invalid colour". To the computer, however, it is as crisp and clear as early morning air! Simply 'assemble'! Using a FOR-NEXT LOOP you will have to rediscover these values, note them and then disassemble, in order to see exactly what happens. The variable 'start' contains the memory adress of the first character after the REM-statement!

Some headerless-programs have too low an adress to be loaded normally. Time for a trick. I will only tell you how, and without an elaborate example, as you are now so at home in cracking -and thus REALLY thinking- that that would be exaggerating. Let's presume that the code starts at adress 24000 and has a length of 6000 bytes. We have to make our own Mcode to load this properly.

```
LD HL, start loaded code (40000)
LD DE, where to (24000)
LD BC, how many bytes (6000)
LDIR
LD SP, 23999
CALL 24000
```


A tip for cat-X-wide: On ducdisc-1 you find "cat3code", needed to get a 3-column CAT on screen. However, an 80 char. printer can print 7 columns wide! This can be altered. Make a copy of the file 'cat3code', load LINK-ED, put the copy into drive 1 and find the file. Byte #0282 (hex) has the number 03. Go into 'OVERWRITE' mode and type a 0 and a 7, folowed by <enter>. Go to another block using key 0, 9, 6 or 7. LINK now asks if the change has to be saved. Answer Y. The file now contains the mcode to give a 7 column CAT. What has happened?

The subroutine LUS2 holds instruction CP 3;(is it now 3?:) This is the number of columns. Using the above method you have changed that into 7. It's that easy. If required, I will provide the new CATcode through the program bank, but only after permission from the author, Marcel van Dongen.

Can someone alter this CAT in such a way that it is sorted, can work out file size in blocks, can print block size of the disc in the first line and also the number of files? It would be absolutely perfect if it could also state the file types and available memory on disc.

An improved CAT like this one is a thing of beauty; even better than a PC with MS-DOS. I have the idea that the IC 6116 holds enough space for this type of utility. I am working on a kind of 'boot' program which initiates the options required and loads the necessary mcode into the 6116. As you see, there are still enough possibilities. Any takers?

Maarten de Haas

=====



Of course we POKE this to a safe, unused adress in memory. (In other words, not at an adress where the program put its own code later on. Make sure the Mcode starts at this routine.

Well, that wasn't hard, was it? If, by chance, things don'r work out, or programs don't work immeadiately, check the stack pointer, the start- and end- adresses, the length and the sequence. Like some wise guy once said, 'Practise makes perfect'

Perhaps it will take you three or four hours to crack a program -certainly in the beginning. Don't give up. Remember, there is nothing as satisfying as having cracked a program using your own wit and charm. Any idiot can push the red button on the MULTI-FACE ONE.

Rob Macare

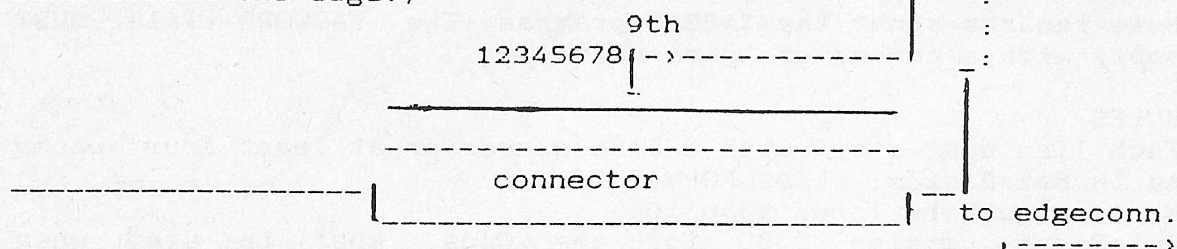
MULTIFACE AND THE OPUS DISCOVERY

By Jurgen Damen

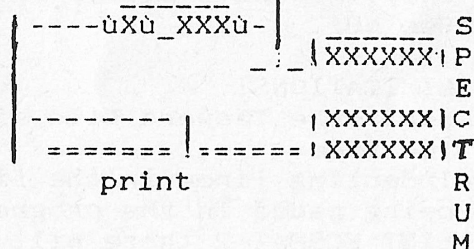
Owners of the Multiface have always been dissatisfied with the fact that it has to be stuck between SPECTRUM & DISCOVERY. That makes it hard to reach the eject button, especially if you have a second 3.5" drive fitted. Many have tried to pass the NMI line through to the side connector, but failed to get it working properly.

Here is the real solution, simple as it is!

Look down at the slot (connector) where the Spectrum connects. Counting from the slot, (after the first 4 pins) cut the 9th pin, solder a wire to the pin on the connector and connect that to the 9th pin on the edge connector (again counting from the slot). Now place a diode between the two ends of the pin you cut through earlier. The side of the diode marked black/yellow should always be on the print-side. (It might also have a Red band around the edge.)



The diode is a general kind, it may be a 1N4148 but all small diodes should be usable. Just for clarity, here is a side view. Do take care (and time) when soldering the print tracks on the edge-connector.



So, essentially, nothing needs to be pulled apart.

Oh, yes! This may be unnecessary, but..... Whatever you do to your computer, always remember to pull the plug before you open it up or try to alter things!!!

Jurgen Damen
Almere

=====

PASTAS AND TASPAS

In case you are unfamiliar with these two terms: PASTAS and TASPAS are programs which allow the user to write a PASCAL program in TASWORD and then transcribe it into PASCAL. At least, that's what TASPAS does. (TAS->PAS = i.e. Tasword->Pascal). Obviously, the program PASTAS does the inverse.

You might not know the program BASTAS as such, but in DUC's 8 and 10 we read about methods of changing a TASWORD file into a BASIC program.

The bastas/tasbas-programs differ a lot from the taspas/pastas programs. Firstly, the taspas/pastas progs were written in machinecode while the tasbas/bastas work in BASIC. Also, the "bas"-programs are based on existing BetaBasic-commando's, unlike in the "pas"-programs. Another difference is, that with TASPAS you may not use line numbers in the text file while with TASBAS that is an absolute MUST. With TASBAS/BASTAS you have less memory available so that you often have to cut progs and files into parts, and MERGE them later.

There is one similiarity:
Remember that a TASWORDfile always uses more memory than a program. There is a solution here: the 'cutting method'. BETA BASIC and PASCAL allow this.

Some remarks about the TASBAS-program. The TASWORD file MUST comply with a number of 'game rules':

RULES:

- Each line must start with a line number or at least four spaces
As in BetaBasics: LIST FORMAT 2.
- No line numbers lower than 10.
- BetaBasic version 3.0D for the Opus, must be used when transcribing. Ordinary Basic-programs can also be transcribed.
- Only TASWORD 3 files can be used. Tasword two users, no panic!
See NB1.

LIMITATIONS:

- Limit the Tasword file size. In BetaBasic you have more than 16k free.
- Indenting lines in the file with spaces results in these also being paced in the program, so with a
LIST FORMAT 2 there will always be more than 2 indentations.
TIP: Indent with one space. This has no visible result in the final program.

CLOSING REMARKS:

In essence it doesn't matter what type of collumns you use. It is advisable to use the same in both BetaBasic and Tasword, however. Make them both 64. There is one snag which I haven't solved: the use of < > >= < > and <= sometimes gives error reports. If this occurs, type: LET b\$=a\$ and GOTO 6. Don't forget to apply the line involved!

What with all the low prices lately, I succumbed to the temptation and bought a 128k Spectrum. The man at the store had no idea whether it would work with the DISCOVERY -which, as it turned out, it did not.

To my great joy I read an article by Rob Macare in one of the DUC's, where he stated that a common cause for my problem was the current stabilizer. I opened the thing up, to look at the plate inside. It turned out to be the correct type, so there was nothing I could do about it anymore. But, as the thing was open anyway, I walked over to my wife and said "Look at all that beautiful electronics for so little money!!"

Her reaction was: "Fine! But what's that centipede doing there?" With a red face I had to admit that what I thought was a brand new invention from Sir Clive, turned out to be a dried up centipede!. After having removed this corpus alienum (strange body), the 128k works like a charm with the DISCOVERY. The proof is this story, written with the 128k.

Now that is what we call a hardware bug!!!

Gertjan Prinsen



SO HERE IT IS...

```

1 CLOSE #4
  LIST FORMAT 2
  INPUT "Name of TW3textfile? "; LINE a$
  OPEN #4;"m";1;a$IN
2 LET b$=""
3 DO WHILE NOT EOF(4)
4   INPUT #4; LINE a$
   IF a$="" THEN GO TO 6
5   IF a$(1 TO 4)="  " THEN
     LET b$=b$+" "+a$(5 TO )
   ELSE KEYIN b$
     LET b$=a$
6   RANDOMIZE CODE INKEY$#4
7 LOOP
8 KEYIN b$
9 CLOSE #4
  LIST 10
  DELETE 1 TO 9
STOP

```

NB1. Tasword 2 users can still use my programs for transferring T3 to T2 and vice versa.

P.Spoelstra

SECTORTEST

For some time after I had bought my DISCOVERY, everything seemed to be going well. Then, every now and then, a disc started backfiring on me. Fortunately, I lived close to the DUC centre in ZOETERMEER, so I joined the club and bought DUCDISC 1. The program SECTORTEST was an interesting program to try out right away, so I could see where my disc went haywire. They all conked out at exactly the same point! So the drive was at fault. I went back to the store and traded my DISCOVERY for one that worked perfectly.

As homage to the program, I rewrote it a bit. It now works twice as fast (no more time to drink your tea) and it looks a bit neater.

So here's the BASIC-listing. The CODEblok needs to be loaded from DUCDISC 1. Good luck.

```
1 REM DISCOVERY BLOCKS TEST                      1985 JOSE PEDRO G.
              1987 MvD Zmeer +60%
5 POKE (PEEK 23613+256*PEEK 23614),205: POKE (PEEK 23613+256*
PEEK 23614+1),254
6 CLS : PRINT INVERSE 1;"                      OPUS-SECTORTEST      "
7 PRINT INVERSE 1;AT 5,0;"1  ALL TRACKS      ";AT 7,0;"2  A
SPECIAL TRACK "
8 LET q$=INKEY$: IF q$<>"1" AND q$<>"2" THEN GO TO 8
9 LET s=-1: LET e=718: IF q$="2" THEN INPUT "TRACK NUMBER (1
TO 40) >";s: LET s=(s-1)*18-1: LET e=s+18: IF s<-1 OR s>718 THEN
GO TO 9
10 LET err=0: IF e>718 THEN LET e=718
15 CLS : PRINT #0;"INSERT DISK THEN PRESS A KEY": GO SUB 200
20 CLS : PRINT INVERSE 1;" OPUS-SECTORTEST      SECTOR:      "
25 PRINT AT 2,0;"ERROR BLOCKS:                  "
30 LET s=s+1: IF s>e THEN GO TO 100
60 LET SH=INT (S/256): POKE 23728,S-SH*256: POKE 23729,SH: LET
SD=USR 65201: PRINT AT 0,29;sd: GO TO 30
100 IF NOT ERR THEN PRINT AT 19,0;"  ALL THE BLOCKS ARE CORREC
T      THE DISK WORKS PERFECTLY                  PRESS A KEY
"
110 IF ERR THEN PRINT AT 21,0;"  ERROR BLOCKS";TAB 19; FLASH
1; " PRESS A KEY "; FLASH 0;AT 21,0;err
120 BEEP .5,30: GO SUB 200: RUN
200 LET q$=INKEY$: IF q$<>"" THEN GO TO 200
205 LET q$=INKEY$: IF q$="" THEN GO TO 205
210 RETURN
500 POKE (PEEK 23613+256*PEEK 23614),4: POKE (PEEK 23613+256*PE
EK 23614+1),19: STOP
1000 IF PEEK 65200<>47 THEN GO TO 500
1010 LET err=err+1: LET x=INT ((err-1)*4/32): PRINT AT x+4,err*4
-x*32-LEN STR$ s;s: BEEP .2,20
1015 IF err>=128 THEN PRINT #0; INVERSE 1;" THIS CAN'T BE A DISC
OVERY DISK ": GO TO 110
1020 GO TO 30
9999 BORDER 0: PAPER 0: INK 7: CLEAR 65199: LOAD *1;"SECTOR  C"
CODE : RUN
```

Martin van Drie

Having read an article about Random Access Files (RAF's) in a previous DUC, I gathered that this subject is not as appreciated as it might be. I hope to improve it's position with this article. Once you know your way about with them, they are wonderfull. I have been using computers for more than ten years now, but this use of the Speccy is still revelation number one!

In the few months I have had my OPUS-SPECCY I have been able to find various utilities to my liking. Many of them use records, for which I use RAF's. One of the utilities is a program which keeps track of competition results and then works out the new pooling (squash, badminton etc.). All participants (123) are recorded with their pool-names and results from previous pools.

Another program is actually a collection of short progs which calculate glazing for ceramics by consulting a record of basic components.

When to use Random Acces files

RAF's are very handy to use when working with data which is not all needed in the computer's memory at the same time. Of course it can also be used when not all the data fits into the memory at the same time. You then load a part of the data, select what you need, and then -as it were- merge it with whatever else you need from successive LOADs. Examples from practise are members administrations (DUC!) client records, collections, etc.

A few terms explained.

Before reaching the core of this article, I will discuss some terms. DATABASE is a common term, and it refers simply to a collection of data. DATABASE MANAGEMENT SYSTEM (DBMS) is a system which takes care of data access (search, addition, alterations, erasures, groups, etc.) A database is built out of RECORDS or lines. Each line contains data about 1 element of the database. (One member out of a complete administration.) Of course you can also keep more than one record for a member, but for simplicity's sake, I will limit myself to one record. Usually a RECORD is subdivided into FIELDS, each representing one piece of info (adress, telephone nu.ber, etc.)

How do I design a database?

Before diving behind your SPECCY you must give it some thought. You must decide what kind of database you wish to create. Let's say a members' administration. How many members must it cope for? Which particulars should it keep for each member? It is obvious that you make a record for each member.

Once you have decided what you wish to store, you can work out how much space you need for each particular, and whether it will be numbers, letters, or a combination of both. (Numerical, character or alphanumerical).

I admit this is more the case when working with the SPECTRUM-OPUS combination, as a database is always treated alpha-numerically; with input # stream;F\$ for example. Once you have determined all these things, you can calculate the record length as the sum of the length of all fields plus 1. That 1 is necessary as the DISCOVERY closes each record with an end-of-record marker (a CR=chr\$ 13). Now you have all you need to

make a RAF.

How do you develop software to cope with this data?

Once again, the best start is to sit down and really think everything through before committing yourself to the keyboard. For example, you will need a list of all the functions. Some obvious ones are: Add New Member, alter address, remove member, list names, addresses and postal codes / countries. Having determined these, it is advisable to design a menu from which each of them is accessible. Each function becomes a subroutine, so that you can always return to the main menu.

Don't forget to include a 'Stop Program' option, as using the BREAK key is a bit messy. Doing it all in this manner will show you that to think it all over before you start programming, is already half the hard work!

A few tips.

1) Keep notes on your records and the layout of the program, so that you will always know what is in each record. You can always use REM statements for this. Don't worry if the REM is 10 lines or longer... it is not a waste of space if it helps you keep track of your programming. It also helps if, at a later stage, you want to alter the program. Always use the same variable for loading and saving, and never use this variable for anything else. Same goes for all the variables of the record-info. If N\$ is a name, then only use it for names! This strictness saves a lot of confusion and frustration when things go wrong.

2) An index is handy if in your records a certain field keeps recurring in different variations. Take for example a members administration. You have decided that it was handy to group members according to the suburbs. You could make an index containing the record numbers of the members in each suburb. This saves a lot of work when searching. In my program, this method increases the search speed by factor 3 to 4. The record numbers in the index of each suburb can be used in a POINT statement. In this way you jump directly to the record needed, and this is the advantage RAF's give us!!!

3) If your records are more intricate, you will undoubtedly work with arrays like DIM P(a,b,c). For each array, the speccy reserves some memory space. This can add up to a lot of K's. If it all fits, you'd say all was O.K. But now the problem: you want to SAVE it all, and using OPUSCAT you see that it takes up 20k. A large part of this will be the variables, which don't need to be SAVED. The remedy is: somewhere in the beginning of your program, you put a line containing only a STOP statement. RUN this line, and only then SAVE.

You will see that the SAVED program is a lot shorter!!! This is because a RUN command erases all variables. And as we do not create new variable using this trick, you clear this area, and therefore it takes up no room on disc. An example: a string array of 10 by 10 and a stringlength of 10 takes up 1k memory. Sometimes this trick can save as much as 13k!

Maarten de Haas

An introduction.....

CALPHY is the name for a routine in the DISCOVERY ROM. (We call this 'D-ROM' on the following pages). The actual function of this routine is controlling peripherals such as the joystick and parallel port as well as the disc drives. Actually the name for this routine is CALL PHYSICAL DEVICE.

It's no wonder that designers of machine code routines for the DISCOVERY orientate themselves around this routine. This routine makes it possible to carry out all the BASIC steps in another manner.

The D-ROM can be paged in by means of a call to address #1708. Because of a bug at this address of the SPECTRUM, the shadow ROM will be activated. Using a slightly higher address - #1748 - the shadow ROM is paged out again.

Between the call to #1708 and #1748 the SPECTRUM ROM (D-ROM) is not in control. The D-ROM is! Therefore we can no longer make use of the S-ROM routines, because it is looking for them in the D-ROM! It is therefore advisable to make use of the D-ROM only when absolutely necessary.

When using the routine with the IC 6116, which takes up the area of the 2k-byte RAM above the D-ROM, but above the S-ROM, take very good care. In this situation it is not possible to page out the D-ROM so here you must use other techniques.

Section 4 explains how to get around this problem. Section 4 is a small booklet containing information about the use of machine code in the DISCOVERY. Explanation of the fact that when paging in, all registers will be SAVED. When paging out all registers will be RESTORED. So, when paging out, registers are not preserved. The use of this situation will be explained later.

The paging in of the D-ROM causes the D-ROM (and possibly the IC 6116) to be stacked in the area of the S-ROM for as far as they overlap each other. Nothing else is changed in the registers, and the stack pointer remains where it was before paging in. So no harm is caused when trying to page the D-ROM after having paged in.

It is, however, harmful when you try to return to the 'old' ROM-situation while the S-ROM is already at hand. The SPECTRUM will give an error-message. Before paging out the D-ROM it is advisable to page in the D-ROM (especially when you've done so much that you've forgotten whether you are dealing with the D-ROM).

ERROR!!!

This is not a reminder that all the above text is incorrect, but a statement of the fact that we will now be handling errors. As it is not possible to deal with all the errors (Murphy lurks all around), we can name the most common, in order to prevent them from happening.

The SPECTRUM recognizes two system-variables which are the most important when RUNNING routines; namely ERR_SP and ERR_NR (respectively 23613/4 and 23610).

When we load in ERR_SP at the address of an error-routine (i.e. it contains a routine which knows what to do when there is an error) then we know that errors will be taken care of. We won't need to worry about error-messages, but can generate them ourselves.

Pressing BREAK could result in printing a line across the bottom of the screen, perhaps a message like 'NOT ALLOWED', while your own routine is continuing as normal. Nevertheless, within your error routine, you must take care that the value in ERR_NR is adjusted, else you might see some very odd messages while in BASIC.

Application.....

Let's apply the routine, aimed at CALPHY. We can use the above perfectly for specific disc operations. For example, to check whether a disc is formatted, or perhaps write-protected, or to check for faulty sectors.

When we start with the possibility to preserve registers in another way than via the stack, we can apply the following method for disc operations. MARK is the address of a byte where the result of a disc operation is stored; later, MARK will be checked to deduce whether disc operation has been carried out correctly or not:

```

CALL          #1708          paged in
*****; here the registers are stored for the operation
LD            (pr1+1),HL      PRESERVE HL
LD            (HL,(ERR_SP))
PUSH          HL              Preserve ERR_SP
HL            HL,f_rout
PUSH          HL              preserve error
                               routine
LD            HL,#1708
PUSH          HL              the option to
                               page in the D-ROM
LD            (ERR)SP,SP      the way to handle
                               the general error
                               routine of the
                               SPECTRUM
pr1 LD          HL,0           restore HL
CALL          CALPHY           things can go
                               wrong
***** if nothing goes wrong we will return from CALPHY
***** without damage.
LD            (pr2+1),HL      preserve HL
POP           HL
POP           HL              clear stack junk
POP           HL
LD            (ERR_SP),HL      restore earlier
                               error-routine
pr2 LD          HL,0           restore HL
***** ; treatment of MARK
CALL          #1748           page out the DROM

```



```

f-rout ; what to do upon error
      POP      HL
      LD        (ERR_SP),HL      restore error
                                   routine
      CALL      1748             page out D-ROM

```

Having outlined the above, we can get down to business.

It is possible to detect whether sectors (or blocks) are still readable (FORMAT? or SECTORTTEST). We can also find out whether we can SAVE a previously LOADED block, and a lot of other things can be discovered as well. Just think of the enormous amount of possibilities!!!

Some more about CALPHY...

CALPHY checks regularly whether BREAK is being pressed. The routine we discussed does that.

Odd things can happen, however, when loading complete blocks using the above method. The routine should report when loading is no good, but it can also occur that you get an unusual BREAK-report.

CALPHY does not take into account that the user can create blocks of 1024 bytes on disc, but NOT take away, for example, 800. The programmer can load 1 to 256 bytes from the block, or the entire block.

A program which makes good and regular use of CALPHY is 'COMBI 2.0'. This is another routine made by Marcel van Dongen, completely in machine code and offering a lot of very handy features.

It directly controls the information present in the IC 6116, if you have it in your OPUS. Oddly enough, there IS a bug in CALPHY. When one changes the info in the disc-tables, and a FORMAT routine is carried out immediately after, the DISCOVERY does not seem to know which is drive 1 and which is drive 2. It seems to have reversed them! THEREFOR THE DISC IN DRIVE 2 IS FORMATTED WHEN THE DISC IN DRIVE 1 IS SUPPOSED TO BE THE VICTIM, AND VICE VERSA!!!

A parallel (or even coinciding) routine works well, but changes nothing in the IC.

I tried to bypass this situation by changing the info in the IC first, and checking whether the disc in drive was formatted. The odd thing is, that when calling the drive through a non-format routine, the bug does not show!

So be warned, if you intend to change the info in the IC 6116.

Rudie Aalders

STOP! READ THIS ADVERTISEMENT.

German Opus User want contact with DUC Members.
If you are interested, please write to me or send a
180 k Disk (tasword 2 text) to the following adress:

(Write English or German please.)

SCIUC
Spectrum-Computer-Info-User-CLub
c/o R.Frank
Carl-Spitzweg-Str.17
D-7920 Heidenheim 5

West Germany

Please state DUC on the letter.

For sale:

- 1) OCP ASSEMBLER / EDITOR g
- 2) OCP FINANCE MANAGER g On tape, 180 column
- 3) OCP ADDRESS MANAGER /
- For '2.50 each
- 4) HURG GAMES DESIGNER
- 5) OCP MACHINE CODE TEST TOOL
- ON TAPE FOR '5
- 6) TRANS EXPRESS '5

Write to Vernon Hampson
57 Bath road
Banbury
Oxon
OX16 0TU
tel: 0295 - 69288

For sale: 10 discettes 3.5" for 35DM
 '14
 Hf140

Write to Jorg Helpenstein
Postfach 100610
4040 Neuss1
WDR

sinclair

30

impuls

RTTY/TX SLOW SCAN GROUP

Up to now reactions to our work group have been limited. There were a few reactions from France and England, and a software offer from PEARSONS COMPUTING.

So all DISCOVERY fans and active DX'ers and OM's: please contact our workgroup!!!

We want to discuss software and hardware for combining our two hobbies, like decoding devices and automatic logbook-keeping programs.

And now to the software offer from PEARSONS COMPUTING. We are offered three different programs on one disc or each of the programs on one disc or cassette. G1FTU CW is a morse code transmit/receive program which works without any interface on the 48k SPECTRUM and SPECTRUM PLUS computers and, as far as I know, on the 128k range with a few snags (I do have more info on that, if you're interested).

G1FTU RTTY does the same thing with RTTY code receiving and transmitting the latest G1FTU SSTV/RX. (The received picture fits neatly on the SPECTRUM screen and the program features a new linearity system which produces good results on both pictures and text displays.)

All programs are sent with complete instructions and there is also an OPUS version, which is most fortunate for us, but up to now only the SSTV mode for complete disc use. In the other two programs the cassette system is still used for loading and saving memory.

The software will be sent to you on one disc: the reduced price for DUC members is not fixed yet, but will be approximately 30 AUSTRIAN schillings plus disc and postage and WITHOUT reduction. For more price details wait for the next news in later DUC mags.

The adress to write to for more details is:
PEARSONS COMPUTING, 42 Chesterfield Road, Barlough, Chesterfield Derbyshire S43 4TT, England.

LATEST INFORMATION**

There is a special club which deals with the problems of radio amateurs and computer fans: THE SINCLAIR AMATEUR RADIO USERS GROUP, c/o Mr. Paul Newman, 3 Red House Lane, Leiston, Suffolk IP16 4JZ, England. Membership in Europe is (as far as I know) 8 AUSTRIAN schillings per year.

That's all for now. Please write to us if you want more info about the group.

Mr. Cyril Dance
Posenergasse 14
A-08045 Graz-Andritz
Austria

KOMIN

S3025 SPECTRUM 128 PLUS 2

De allernieuwste computer. Ingebouwde datarecorder, 2 joystickpoorten, rs 232 interface, 128k Ram en 32k Rom, midpoort, keypadpoort, RGBpoort en een professioneel toetsenbord. Bovendien Spectrum 48k compatible.

f 495,00

In deze advertentie
is slechts een
keuze gemaakt uit
het grote assortiment artikelen
dat wij voor
de Spectrum 48/124/128 + 2 kunnen
leveren.

Komin
is de officiële
importeur van

CST *Thor*

en

SANDY

S3000 SPECTRUM 128k

f 345,00

De laatste computer die nog door Sinclair zelf is geproduceerd. Interface 1 compatible. Haast u, nu nog in beperkte aantallen te koop.

T740 SAGA 2001

f 499,00

Het paradepaardje van SAGA. Fantastisch toetsenbord. Infrarood afstandsbesturing. Special-design Qwerty-toetsenbord, numeriek veld en 40 functietoetsen, waarvan 20 met 4-dubbele functie.

T720 MUSIC MACHINE

f 199,00

Soundsampler voor uw Hifi-stereoinstallatie. Snel te menugestuurde scherm invoer. Elk extern geluid kan worden opgenomen, veranderd en afgespeeld via het toetsenbord. Ingebouwde drumgenerator, voorgeprogrammeerde ritmes. Engelstalige handleiding, microfoon en demonstratietape.

T730 RAMPRINT

f 175,00

Centronics parallelinterface met ingebouwde tekstverwerker in Rom, dus onmiddellijk beschikbaar. Eenvoudige commando's. Ingebouwde joystickpoort. Inclusief Centronics printerkabel en engelse handleiding.

K830 THE WRITER

f 79,00

Professioneel, maar toch zeer gebruiksvriendelijke tekstverwerker. Regels tot max. 127 tekens, printer spooler, muis of joystickbesturing en pull-down menu's. Tasword Two, Quill en Wordstar bestanden kunnen moeiteloos worden ingelezen. Bovendien is communicatie mogelijk via rs232 met praktisch iedere microcomputer.

K840 OCP ADVANCED ARTSTUDIO

f 119,00

Grandioos tekenprogramma met muisbesturingssoftware.

Bij ons vindt u de nieuwste spelsoftware voor de Spectrum 48/124/128 + 2 computers.

alle prijzen zijn inclusief b.t.w.

KOMIN COMPUTERSYSTEMEN

Postbus 1805, 5602 CA Eindhoven, De Greefstraat 15A, 5622 GJ Eindhoven
Telefoon 040 - 456660*, Telex 59032 Notel NL.
Bank: Rabo 15.90.05.701, A.B.N. 52.82.75.615, Giro: 46.43.401.

**NEDERLANDS GROOTSTE
sinclair SPECIALIST**

Verzending
onder
rembours of
bij
vooruitbetaling.
Wijzigingen
voorbehouden.

Informatie
en bestellen
kan ook
telefonisch
tussen

10.00 en 16.00 uur
telefoon 040 - 456660

VRAAG ONZE GRATIS CATALOGUS

ONZE SHOWROOM IS GEOPEND MAANDAG T/M VRIJDAG VAN 9.00 TOT 17.00 UUR